Adaptive user interfaces for health care applications

Krish Ramachandran

January 20, 2009

Adaptive user interfaces assist users in accomplishing tasks in an application and construct a model of the user's preferences so as to serve them better in the future. Examples include systems that filter news stories, recommend products, and so on. This approach to personalized services is relatively new but has great potential for improving the effectiveness of human-computer interfaces. Health care is a significant area where adaptive user interfaces can be of great use. Health care users range from having little computer knowledge (for example, some nurses or doctors) to having expert computer knowledge (for example, system administrators). And, there can be many other distinguishing factors when it comes to patients. Therefore, adapting a computer application's interface to different types of users is important to improve the usability of such applications. Two major techniques used for adaptation are adaptive presentation and adaptive navigation. Adaptive presentation involves personalizing the contents presented to the user. Adaptive navigation involves customizing ways by which users complete their tasks in the application. These techniques can be used to enhance the usability of health care applications, thereby contributing to their success.

Introduction

Web applications are becoming a standard way to provide and manage services in different areas. Replacing complex manual tasks with computer applications causes an increase in the complexity of the user interfaces of such applications. A good user interface facilitates effective communication between the user and the software. Good design of such an interface is vital for the success of any computer application. As a result of this increasing complexity, creators of Web applications are posed with big challenges to simplify the user interfaces and help users complete their tasks with ease. Such simplification is challenging considering the diversity of users in terms of age, geography, knowledge level, and so on. An interface that is simple for one user might not be simple for another. Therefore, it is important to adapt the user interface to different types of users based on their profiles.

Adaptive user interfaces also make it easier to comply with the changing rules and regulations around access control of information. They provide more flexibility to administrators in dealing with such changes. Such adaptive interfaces can be used in a wide range of domains and in a variety of applications. This article explores the two major techniques used to create adaptive interfaces and provides examples of each.
Techniques for adaptive interfaces

The two major techniques used for adaptive interfaces are:

1. Adaptive presentation
2. Adaptive navigation

The following sections explain each technique in more detail.

Adaptive presentation

The goal of adaptive presentation is to adapt the content of a page to the current knowledge, goals, and other significant characteristics of the user. For example, a competent user can understand more detailed information, whereas a novice user requires additional explanations. In most Web applications, the contents of a page may not only be text, but might also include multimedia items. For this reason, adaptive presentation techniques must focus on multimedia items such as images, videos, and audio, in addition to text.

Adaptive presentation can be implemented in many ways. One way is to hide some of the content in the page that is not relevant to the user’s level of knowledge about the concept. For example, low-level details on the page can be hidden from users with little or basic knowledge of the concept since they may not be ready for that level of detail. On the other hand, additional explanations usually required by novice users can be hidden from expert users who don’t need those explanations. Another method is to provide different information and links on a particular page based on the user’s knowledge of related concepts.

Examples of adaptive presentation

Consider the example of creating a patient profile for a specific treatment. If the user is a doctor or health care expert, the user interface can display advanced medical details that he or she can enter or obtain from the patient. However, if the user is a system administrator with limited domain knowledge, those fields might not make sense and can be hidden. Such hidden fields can then be displayed later to the doctor or health care expert, who can complete the additional required details. Another example of adaptive presentation is the language used to explain different elements in a page. For a doctor or health care expert, language can use advanced medical terminologies to give more information, whereas for a user with limited domain knowledge, language can be simple to give basic information.

Listing 1 shows a simple XML code sample for adaptive presentation.

Listing 1 - Adaptive presentation in XML code

```xml
<input name = 'anesthesia level'>
  <for user = 'doctor' visible = 'true' /> 
  <for user = 'computer administrator' visible = 'false'/>
</input>

<label name = 'anesthesia description'>
  <for user = 'doctor' showAdvancedText = 'true' />
  <for user = 'computer administrator' showAdvancedText = 'false'/>
</label>
```
In the above examples, the input 'anesthesia level' is shown only to doctors and not administrators, using the Boolean flag `visible`. The label 'anesthesia description' can be shown with an advanced description to the doctor and with a basic description to the administrator, using the flag `showAdvancedText`. These examples demonstrate adaptive presentation using XML at the user interface level, and such adaptations can also be part of the logical tier of the application. The type of implementation depends on the business and environmental requirements of the application.

**Implementation view of adaptive presentation**

*Figure 1. Simplified adaptive presentation implementation*

Adaptive presentation can be implemented using a set of rules and a rule interpreter that identifies the appropriate rule to be used for particular content, based on the user profile. Each rule corresponds to a specific presentation strategy which corresponds to one or more UI elements. The contents in the page can be displayed to the user after applying the set of rules to the concept represented by the page. Rules can generally be stored as files or in relational databases and can be modified. The rule engine can be implemented in the processing module of the application and would contain logic to select a rule based on the user profile and page concept. A presentation module would trigger a rule, or a set of rules, based on the current page and adapt the contents to comply with the rules associated with the page.

**Adaptive navigation**

The goal of adaptive navigation is to help users find the path to accomplish a specific goal in the application. It involves adapting the way navigational links are presented to users based on their current goal, knowledge about the task, and other relevant characteristics. Users with diverse goals would require different navigational support to accomplish their tasks. For example, a qualified user might not require much help in finding relevant information on a page, whereas a novice user might require more navigational aids to find pertinent information on the same page.

Adaptive navigation can be implemented in many ways. It can be used to provide guidance to a user in finding the required information, or to provide direction identifying the user's position in the page, or to help manage personalized views on information spaces. Guidance can be provided adaptively to assist the user in identifying the shortest path to complete a specific task. If the user's goal is known, even partially, that information can be used to provide personalized guidance. Such guidance can also assist the user in making one navigational step by suggesting the most relevant...
Examples of adaptive navigation

Consider the example of creating a patient profile. System administrators might like to complete the task of creating a patient profile quickly by combining two or three subtasks on a single page, whereas a doctor or health care expert with limited computer knowledge might like to go through the steps one by one, similar to a wizard. Another example of adaptive navigation is customizing the links displayed on the page such that domain experts would access them in a different order than a user with limited domain knowledge. Some links can also be hidden or might point to different tasks based on the user type.

Listing 2. A simple XML code sample for adaptive navigation

```xml
<section name = 'blood tests section'>
  <for user = 'doctor' showAllTests = 'true' showTestsInTabs = 'false'/>
  <for user = 'administrator' showAllTests = 'false' showTestsInTabs = 'true'/>
</section>

<link name = 'blood tests description'>
  <for user = 'doctor' visible = 'false' />
  <for user = 'administrator' visible = 'true'/>
</link>
```

In the above example (Listing 2), the section 'blood tests' can be shown as a single page to the administrators and as a set of tabs to the doctors, using Boolean flags `showAllTests` and `showTestsInTabs`. The link 'blood tests description', describing the blood tests entered, can be shown to the administrator and not to the doctor, using the flag 'visible'. These examples demonstrate adaptive navigation using XML at the user interface level. Such adaptive navigation can also be part of the processing module or middle tier of the software application.

Implementation view of adaptive navigation

Figure 2. Simplified adaptive navigation implementation

Adaptive navigation can be implemented using a set of rules, with each rule corresponding to one or more navigational elements in the user interface. The rule engine will have rules corresponding
to the actions to be taken on a specific link, or a set of links, based on the scenario and the user profile. The navigational links in a page will be displayed after processing a specific set of rules with respect to the concept represented by the page and the user's profile. Rules can generally be stored in files or in relational databases and can be modified. The rule engine will be implemented in the processing module of the application and would contain the set of rules associated with displaying a specific link or set of links on a page with respect to the user’s background, level of knowledge, and any other important information pertaining to the application. The presentation module would trigger a rule or a set of rules based on the current active page in the application and adapt the navigational links of that page to the user based on such rules.

Solution implementation

Figure 3. Simplified adaptive system architecture

The following list provides an explanation of the various components of the architecture:

- **Database:**
  Rules databases store the rules associated with adaptive presentation and adaptive navigation. Rules can be stored in a relational database or as XML files, depending on the design of the system. Such rules must be implemented keeping in mind that they might be updated or modified in the future. Basically, a profile of the user is stored, and that profile is used to provide adaptation. A user profile consists of personal details about the user, knowledge of the user on domain concepts, user goals, and any other information that is required by the application. In terms of implementation, this model will most likely exist as one or more tables in the database. For example, user information could consist of age, expertise in health care, level of computer knowledge, education, and so on. Any relational database can be used to store user information.

- **Adaptation engine:**
  The adaptation engine consists of an adaptive presentation engine, an adaptive navigation engine, and a business logic interpreter to make sure that the adaptation conforms to the business rules of the application. The adaptation rules for presentation and navigation will be implemented in the processing module of the application using a suitable programming language or a middle tier framework. A rule engine can be implemented using a set of rules as part of the program itself, or can be stored as files using a database. A rule interpreter will be used to resolve a particular rule for a specific scenario and user profile.

- **Adaptive user interface:**
A user interface will typically be composed of different variations of presentation and navigational elements that can be adapted to different users. The UI framework used to implement adaptive presentation and adaptive navigation must be flexible enough to represent various elements of the interface in ways suitable for adaptation.

Conclusion

Adaptive user interfaces are a new and quickly developing aspect of software applications. The main characteristics are the adaptation techniques used to personalize the interface to a particular user. Adaptive presentation assists in personalizing the contents to the user, and adaptive navigation guides the user through the tasks towards completion. Adaptive user interfaces would bring great value to the health care domain, as the knowledge of IT in health care is sometimes low. Such adaptations differ, based on the goals and business context, and would significantly improve the usability of health care applications.